



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**PEMILAH BARANG MENGGUNAKAN DOBOT MAGICIAN
DAN *ROBOT OPERATING SYSTEM (ROS)* DENGAN
ALGORITMA *HOMOGRAPHY*
OPEN CV**

SKRIPSI

**POLITEKNIK
NEGERI
JAKARTA**

Donizetta Daffa Aqilla Harley

2003431024

PROGRAM STUDI INSTRUMENTASI DAN KONTROL

INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2024



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**PEMILAH BARANG MENGGUNAKAN DOBOT MAGICIAN
DAN *ROBOT OPERATING SYSTEM (ROS)* DENGAN
ALGORITMA *HOMOGRAPHY*
OPEN CV**

SKRIPSI

Diajukan sebagai salah satu syarat untuk mendapatkan gelar

Sarjana Terapan

Donizetta Daffa Aqilla Harley

2003431024

PROGRAM STUDI INSTRUMENTASI DAN KONTROL

INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2024



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan satu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Donizetta Daffa Aqilla Harley

NIM : 2003431024

Tanda Tangan :

Tanggal : 21 Agustus 2024

**POLITEKNIK
NEGERI
JAKARTA**



LEMBAR PENGESAHAN SKRIPSI

Tugas akhir diajukan oleh:

Nama : Donizetta Daffa Aqilla Harley
NIM : 2003431024
Program Studi : Instrumentasi dan Kontrol Industri
Judul Tugas Akhir : Pemilah Barang Menggunakan Dobot Magician dan *Robot Operating System* (ROS) dengan Algoritma *homography* Open CV

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 21 Agustus dan dinyatakan LULUS.

Pembimbing : Hariyanto, S.Pd., M.T.
NIP. 199101282020121008

Depok, 21 Agustus 2024

Disahkan oleh

Ketua Jurusan Teknik Elektro

Dr. Murni Dwiyanti, S.T., M.T.
NIP. 197803312003122002

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Terapan Politeknik. Tugas Akhir ini berjudul “**Pemilah Barang Menggunakan Dobot Magician dan Robot Operating System (ROS) dengan Algoritma homography Open CV**”

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Dr. Murie Dwiyanti, S.T., M.T. Selaku Ketua Jurusan Teknik Elektro;
2. Sulis Setiowati, S.Pd., Eng Selaku Kepala Program Studi dan Dosen Instrumentasi dan Kontrol Industri;
3. Hariyanto, S.Pd., M,T selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan tugas akhir ini;
4. Pihak rekan kerja yang telah banyak membantu dalam usaha memperoleh data dan membantu dalam kesulitan yang ada;
5. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral; dan
6. Beberapa Teman yang saya tidak sebutkan satu per satu yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 21 Agustus 2024

Penulis

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Abstrak

Sistem pemilahan barang otomatis merupakan teknologi penting dalam industri untuk meningkatkan efisiensi dan akurasi produksi. Penelitian ini bertujuan untuk mengembangkan dan menguji sistem pemilah barang menggunakan lengan robotik Dobot Magician yang terintegrasi dengan Robot Operating System (ROS) dan algoritma homografi dari OpenCV. Penelitian dilakukan dengan memanfaatkan kamera Logitech C270 untuk deteksi objek dan kalibrasi koordinat. Metode yang digunakan melibatkan proses kalibrasi kamera dan lengan robot untuk memastikan akurasi dalam pemindahan objek. Hasil pengujian menunjukkan bahwa sistem ini memiliki faktor skala sebesar 0,41085 pada sumbu X dan 0,4802 pada sumbu Y, dengan keberhasilan deteksi dan pemindahan objek mencapai 90%. Sistem ini terbukti efektif, meskipun masih menghadapi tantangan dalam karakteristik tidak linear yang mempengaruhi prediksi pergerakan di lingkungan kerja yang lebih kompleks. Hasil penelitian ini menunjukkan potensi besar untuk penerapan sistem ini dalam industri, dengan pengembangan lebih lanjut yang dapat meningkatkan akurasi dan efisiensi operasional.

Kata kunci: Dobot Magician, ROS, homografi, kalibrasi, efisiensi

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Abstract

An automated sorting system is a crucial technology in industries to enhance production efficiency and accuracy. This research aims to develop and test a sorting system using the Dobot Magician robotic arm integrated with the Robot Operating System (ROS) and a homography algorithm from OpenCV. The study employs a Logitech C270 camera for object detection and coordinate calibration. The method involves camera and robotic arm calibration processes to ensure accuracy in object handling and movement. The results show that the system has a scale factor of 0.41085 on the X-axis and 0.4802 on the Y-axis, with an 85% success rate in object detection and transfer. The system proves to be effective, although it still faces challenges related to the non-linear characteristics that affect movement predictions in more complex work environments. The findings indicate significant potential for applying this system in industrial settings, with further developments needed to enhance operational accuracy and efficiency.

Keywords: *Dobot Magician, ROS, homography, calibration, efficiency*

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
LEMBAR PENGESAHAN SKRIPSI	iv
KATA PENGANTAR	v
Abstrak	vi
Abstract	vii
DAFTAR ISI.....	viii
Daftar Tabel	x
Daftar Gambar.....	xi
Daftar Lampiran	xii
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	3
1.3. Tujuan.....	3
1.4. Luaran.....	3
BAB II.....	4
TINJAUAN PUSTAKA	4
2.1 <i>State of the art</i>	4
2.2 <i>Dobot Magician</i>	9
2.3. <i>Robot Operating System (ROS)</i>	11
2.4. Linux dan Ubuntu.....	11
2.5. Visual Studio Code.....	12
2.6. <i>Python</i>	12
2.7. <i>Algoritma homography</i>	13
2.8. Kalibrasi Kamera.....	13
2.9. Lampu Baseus Magnetic	14
2.10. Kamera <i>Webcam</i>	14
BAB III	15
PERANCANGAN DAN REALISASI	15
3.1. Rancangan Alat	15
3.1.1. Deskripsi Alat	15

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.1.2.	Cara Kerja Alat	16
3.1.3.	Spesifikasi Alat	18
3.1.4.	Diagram Blok	20
3.2.	Realisasi Alat.....	21
3.2.1.	Realisasi Rancang Bangun Alat	21
3.2.2.	User Interface Sistem	22
3.2.3.	Realisasi Program.....	24
3.2.4.	Proses Integrasi	29
BAB IV	33
PEMBAHASAN	33
4.1.	Deteksi Objek berdasarkan Homography.....	33
4.1.1.	Deskripsi pengujian.....	33
4.1.2.	Prosedur Pengujian	33
4.1.3.	Data Hasil Pengujian.....	35
4.2.	Pemetaan <i>Workspace</i>	39
4.2.1.	Deskripsi pengujian.....	39
4.2.2.	Prosedur Pengujian	40
4.2.3.	Data Hasil Pengujian.....	41
4.2.4.	Analisis Data	44
BAB V	61
PENUTUP	61
5.1.	Kesimpulan.....	61
5.2.	Saran	61
DAFTAR PUSTAKA	63
LAMPIRAN	xiii



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Daftar Tabel

Tabel 3. 1 Spesifikasi Hardware yang digunakan.....	18
Tabel 3. 2 Spesifikasi Mekanik Alat.....	19
Tabel 3. 3 Keterangan Gambar 3.3	22
Tabel 3. 4 keterangan Gambar 3.4	22
Tabel 4. 1 Alat dan Bahan Uji Coba.....	40
Tabel 4. 2 Data Hasil Kalibrasi Workspace.....	41





Daftar Gambar

Gambar 2.1 Dobot Magician.....	9
Gambar 2.2 Jangkauan Dobot Magic Panjang Radius 320 mm.....	10
Gambar 2.3 Skema Pemasangan Pompa angin atau air pump dengan daya 12V dari port SW1.....	10
Gambar 2.4 Blok Arsitektur ROS.....	11
Gambar 3.1 Flowchart Perancangan Alat.....	15
Gambar 3.2 Flowchart Cara Kerja Alat.....	17
Gambar 3.3 Blok Diagram Sistem.....	20
Gambar 3.4 Tampak Depan Alat.....	21
Gambar 3.5 Tampak Belakang Alat.....	21
Gambar 3.6 UI Sistem.....	23
Gambar 4.1 Pengujian benda pertama.....	36
Gambar 4.2 Pengujian benda kedua.....	37
Gambar 4.3 Hasil Pemetaan Kamera dan Lengan Robot.....	44
Gambar 4.4 Grafik Hasil Pengujian Kalibrasi Workspace.....	44

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Daftar Lampiran

Lampiran 1. Daftar Riwayat.....	xiii
Lampiran 2. Datasheet Dobot Magician	xiv
Lampiran 3. Dokumentasi Pengujian Alat	xv
Lampiran 4. Realisasi Program Pengambilan Dan Klasifikasi	xvi
Lampiran 5. Realisasi Program PythonUI	xxxi
Lampiran 6. Realisasi Program Homografi	xlix
Lampiran 7. Link Program Pemetaan Dobot Magician	liii
Lampiran 8. Link Program homography Dobot Magician.....	liv
Lampiran 9. Alat bantu Kalibrasi Kamera	lv





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I

PENDAHULUAN

1.1. Latar Belakang

Revolusi industri keempat, yang sering kali ditandai oleh teknologi yang menghapus batas antara bidang fisik, digital, dan biologis, secara fundamental mengubah wajah pekerjaan banyak sektor. Integrasi teknologi canggih dan otomatisasi mendorong efisiensi, produktivitas, dan inovasi dengan laju yang belum pernah terjadi sebelumnya. Sektor-sektor kunci, termasuk logistik dan layanan pengiriman, akan menyaksikan perubahan transformatif dengan integrasi tenaga kerja robotik, khususnya melalui penggunaan lengan robot yang canggih. Penelitian ini untuk mengetahui peran penting lengan robot otomatis dalam meningkatkan efisiensi operasional melalui deteksi dan penanganan objek yang presisi, memanfaatkan teknologi seperti *Monocular Webcam Cameras* dan pemrograman berarsitektur homografi untuk pengenalan objek dengan beberapa tahap dan peran yang berbeda.

Pertama, lengan robot yang dulunya primitif dan terbatas pada tugas-tugas repetitif dasar, telah berkembang menjadi aset multifungsi yang rumit yang mampu melakukan berbagai operasi. Tangan robotik ini dirancang untuk melakukan tugas-tugas yang memerlukan presisi dan ketangkasan, seperti mengangkat objek dengan berbagai bentuk, berat, dan ukuran. Kemajuan signifikan dalam bidang ini adalah kemunculan lengan robot yang dilengkapi dengan sensor canggih dan kemampuan pengenalan objek, memungkinkan otomatisasi yang lebih halus dan efektif (Afrisal, H., Soesanti, I., & Cahyadi, A. I. (2019). *Estimasi Posisi Dengan Menggunakan Kamera Monokular. Transmisi: Jurnal Ilmiah Teknik Elektro*, 21(1), 1-9).

Kedua, *monocular Webcam Cameras* muncul sebagai teknologi penting dalam konteks otomatisasi robotik. Webcam berfungsi sebagai sensor canggih untuk deteksi objek. Menurut penelitian oleh Ali Basrah Zhafranul Nafis (2021), kamera-kamera ini dapat dengan akurat mengidentifikasi objek berdasarkan parameter seperti ukuran, bentuk, dan warna melalui pemrosesan

gambar multi-tahap. *Monocular Webcam Camera* beroperasi dengan menangkap gambar yang kemudian dianalisis menggunakan serangkaian algoritma untuk menentukan karakteristik objek(Pulungan, A. B., & Nafis, Z. (2021). Rancangan Alat Pendeteksi Benda dengan Berdasarkan Warna, Bentuk, dan Ukuran dengan Webcam. (*JTEIN: Jurnal Teknik Elektro Indonesia*, 2 (1), 49-54) Tingkat deteksi yang halus ini sangat penting untuk otomatisasi di berbagai sektor.

Ketiga, pemrosesan gambar dan pemrograman berarsitektur homografi untuk Pengenalan Objek. Untuk memanfaatkan potensi *Monocular Webcam Cameras*, pemrograman Arsitektur Homografi dapat digunakan untuk mengembangkan sistem pengenalan objek yang kuat. Arsitektur Homografi dengan pustaka yang kuat dan kemudahan penggunaannya, sangat cocok untuk tugas-tugas yang melibatkan penglihatan komputer dan pembelajaran mesin. Pustaka seperti OpenCV dapat digunakan untuk memproses gambar dan melatih model untuk mengenali atribut tertentu. Dengan membuat program berarsitektur homografi yang mengidentifikasi dan mengklasifikasikan objek berdasarkan kriteria yang telah ditentukan, sistem ini dapat diintegrasikan dengan lengan robotik secara mulus.

Sebagai contoh, dalam pengaturan logistik, lengan robotik yang dilengkapi dengan sistem deteksi ini dapat diprogram untuk mengenali dan menyortir benda dengan bentuk kotak. Demikian pula, dalam layanan pengiriman, lengan robotik ini dapat mengategorikan paket, memastikan bahwa setiap paket ditangani sesuai dengan spesifikasi bentuk dan beratnya. Integrasi *Monocular Webcam Cameras* dan pengenalan objek berbasis arsitektur homografi meningkatkan kemampuan lengan robotik untuk membedakan dan memanipulasi objek dengan tingkat akurasi dan efisiensi yang tinggi.

Pemanfaatan lengan robotik dengan kemampuan pengenalan benda berupa objek kotak yang canggih memiliki implikasi mendalam bagi berbagai industri. Dalam layanan pengiriman, teknologi ini dapat merampingkan proses penyortiran dan pengemasan, mempercepat waktu pengiriman dan mengurangi

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



kesalahan. Di luar sektor-sektor ini, aplikasi teknologi semacam ini sangat luas, termasuk manufaktur, perawatan kesehatan, dan ritel.

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas, dapat diperoleh perumusan masalah sebagai berikut:

- A. Bagaimana membuat sistem pemilah benda Otomatis Dengan Dobot Magician menggunakan Arsitektur homografi?
- B. Bagaimana mengimplementasikan atau mengintegrasikan sistem *Machine Vision* dengan tangan robot *Dobot Magician*?

1.3. Tujuan

Adapun Tujuan yang ingin dicapai yaitu :

- A. Mampu merancang mesin sortir benda dengan *dobot magician* dan Kamera *Monokuler Webcam*
- B. Mampu mengintegrasikan sistem pemilah benda dengan menggunakan *machine vision* berarsitektur homografi.

1.4. Luaran

Luaran yang diharapkan dari penelitian ini yaitu:

- A. Laporan tugas akhir yang menyediakan informasi dan inovatif dalam hal sistem pemilah benda menggunakan *machine vision*
- B. Dapat menganalisis pengimplementasian sistem *machine vision* terhadap mesin pemilah benda menggunakan robot *Dobot Magician*.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan pembahasan dan data analisa dari hasil pengujian yang sudah dilakukan, penulis dapat menyimpulkan sebagai berikut :

1. Membuat sistem Pemilah Barang Otomatis Dengan Dobot Magician menggunakan Arsitektur homografi dengan cara mendeteksi dan mengangkat objek dengan akurat menggunakan teknik homografi, memungkinkan pemetaan titik-titik gambar menjadi koordinat robot secara tepat melalui integrasi dengan *Robot Operating System* (ROS)
2. Mengimplementasikan atau mengintegrasikan sistem *Machine Vision* dengan tangan robot *Dobot Magician* dengan cara melakukan uji coba tingkat akurasi dengan melakukan proses pemetaan dan algoritma scalling.
3. Berdasarkan hasil Pengujian menunjukkan bahwa sistem dapat mendeteksi dan memindahkan objek dengan akurasi memadai untuk "mengambil objek kotak," dengan faktor skala sebesar 0,41085 pada sumbu X dan 0,4802 pada sumbu Y dan setelah mencoba beberapa metode mendapatkan hasil akurat, hasil percobaan ini bahwa, karakteristik didapatkan dari sistem ini adalah tidak linear.

5.2. Saran

Setelah melakukan penelitian dan analisis oleh penulis, adapun beberapa saran untuk penelitian selanjutnya adalah sebagai berikut :

1. Sistem ini dapat dikembangkan lebih lanjut dengan menerapkan metode kontrol seperti kinematika maju (*forward kinematics*) dan kinematika terbalik (*inverse kinematics*) untuk kontrol sistem. Selain itu, penggunaan dapat ditingkatkan melalui pengembangan

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



perangkat lunak antarmuka pengguna (UI) atau simulasi pergerakan menggunakan aplikasi seperti *Rviz* atau *Coppeliassim*.

2. Disarankan untuk mengintegrasikan sistem yang telah dikembangkan ke dalam lingkungan industri nyata atau skala produksi kecil guna mengevaluasi performa sistem dalam kondisi kerja lebih dinamis dan berkembang. Implementasi dalam konteks seperti konveyor belt, lini perakitan, atau penggunaan alat *end-effector* khusus dapat memberikan wawasan yang lebih mendalam.
3. Kalibrasi sistem perlu ditingkatkan untuk mengakomodasi variasi dalam ukuran dan bentuk objek yang lebih luas. Selain itu, pengembangan sistem yang lebih skalabel untuk berbagai jenis objek akan meningkatkan aplikasi penelitian ini di berbagai industri.

© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Afrisal, H., Soesanti, I., & Cahyadi, A. I. (2019). Estimasi posisi dengan menggunakan kamera monokular. *Transmisi: Jurnal Ilmiah Teknik Elektro*, 21(1), 1–9.
- Dobot. (2024). *Dobot Magician Educative Industrial Robot*. https://robotdobot.com/catalog_list/roboty/Magician_datasheet_en.pdf (Diakses pada 25 Agustus 2024).
- Nuralam, D. (2020). Pemodelan machine vision untuk pemilah kualitas telur bebek pada skala industri. Dalam *Seminar Nasional Teknik Elektro* (hlm. 127–131).
- OpenCV. (2021). *OpenCV: Basic Concepts of the Homography Explained with Code*. https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html?ref=blog.roboflow.com (Diakses pada 25 Agustus 2024).
- Pulungan, A. B., & Nafis, Z. (2021). Rancangan alat pendeteksi benda berdasarkan warna, bentuk, dan ukuran dengan webcam. *JTEIN: Jurnal Teknik Elektro Indonesia*, 2(1), 49–54.
- ROS.org. (2021). *ROS Tutorials*. <https://www.ros.org/> (Diakses pada 25 Agustus 2024).
- Mallick, S. (2021). *Homography Examples Using OpenCV (Python/C++)*. <https://learnopencv.com> (Diakses pada 25 Agustus 2024).
- Sukusvieri, A. (2020). *TA: Implementasi Metode Single Shot Detector (SSD) Untuk Pengenalan Wajah* (Disertasi PhD). Universitas Dinamika. <https://repository.dinamika.ac.id/id/eprint/4215/> (Diakses pada 25 Agustus 2024).
- Wang, R., Guo, X., Li, S., & Wang, L. (2023). Separation and calibration method of structural parameters of 6R tandem robotic arm based on binocular vision. *Mathematics*, 11(11), 2491.
- Yunanto, T. C., Gunadi, K., & Purbowo, A. N. (2020). Implementasi convolutional neural network untuk mengetahui buah tomat yang matang pada pohon tomat menggunakan perangkat Android. *Jurnal Infra*, 8(1), 306–311.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1. Daftar Riwayat



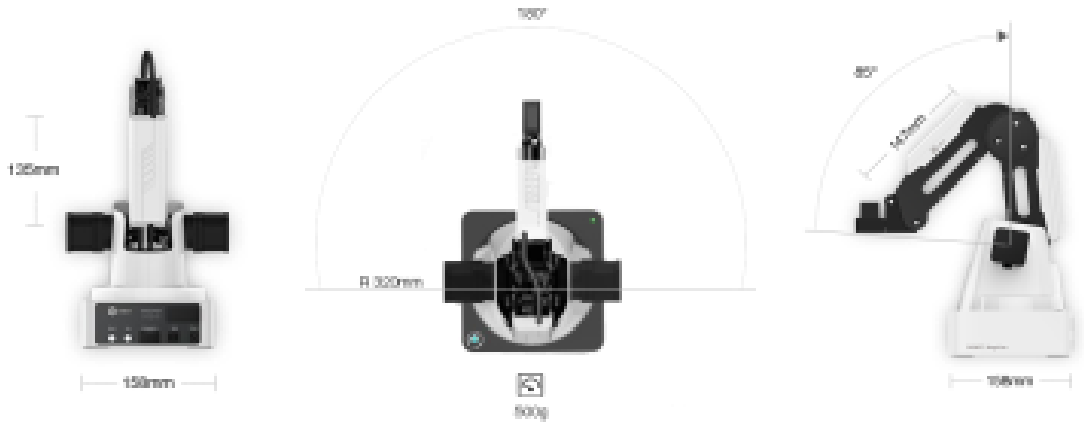
Nama : Donizetta Daffa Aqilla Harley

Lahir di Palu 17 Mei 2002, penulis merupakan anak pertama dari 2 bersaudara. Lulus dari SDN 008 Tarakan Kalimantan Utara pada tahun 2014 , SMPIT Al-Muhajirin Depok pada tahun 2017 dan SMK Penerbangan Atang Senjaya Bogor pada tahun 2020. Kemudian Melanjutkan Kuliah dengan Program Studi Instrumentasi dan Kontrol Industri, Teknik Elektro di Politeknik Negeri Jakarta dari tahun 2020-2024. Mengikuti Program Indonesian Internasional Student Mobility awards (IISMA) di semester 7 di The Haque University of Applied Sciences-Netherlands



**POLITEKNIK
NEGERI
JAKARTA**

Lampiran 2. Datasheet Dobot Magician



DOBOT Magician Specifications

Degree of freedom	4	
Maximum load	500 g	
Maximum reach	320 mm	
Repeatability	0,2 mm	
Communication Interface	USB (Wi-Fi, Bluetooth)	
Voltage	100~240 V, 50~60 Hz	
Power Input	12 V, 6,5 A	
Power	60 W	
Working temperature	-10°C to +60°C	
Software	DobotStudio, DobotBlockly, Repeller Host,	
SDK (Software Development Kit)	Dobot Program Library, Communication Protocol	
Extensible I/O Interface	10x I/O 4x output 12 V 2x stepper motor Interface	
Axis 1 - Base	-90° to 90°	320°/s
Axis 2 - Rear arm	0° to 85°	320°/s
Axis 3 - Front arm	-10° to 95°	320°/s
Axis 4 - Tool rotation	-90° to 90°	480°/s
Weight	3,4 kg	
Weight of package (Standard Version)	7,2 kg	
Weight of package (Educational Version)	8,0 kg	
Material	Aluminum Alloy 6061, ABS Engineering Plastic	
Rozměr balení (D x Š x V)	380 mm x 385 mm x 480 mm	

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

tanpa izin Politeknik Negeri Jakarta

Lampiran 3. Dokumentasi Pengujian Alat

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 4. Realisasi Program Pengambilan Dan Klasifikasi

```
import rospy
from dobot.srv import SetPTPCmd, SetPTPCmdRequest
from dobot.srv import SetEndEffectorSuctionCup, SetEndEffectorSuctionCupRequest
import cv2 as cv
import numpy as np
import tkinter as tk
from tkinter import Label, Entry
import threading
import time

# Global variables
points = []
tracking = False
initial_gray = None
frame = None
homography_matrix = None
current_centroid = None
last_update_time = time.time()
scale_factor = 0.314 # New scale factor applied to both axes

# Updated Robot coordinate limits for the pickup area
X_MIN, X_MAX = -135, 280 # X limits remain the same
Y_MIN, Y_MAX = 55, 260 # Updated Y limits based on your robot's pickup area

# Camera matrix and distortion coefficients (using your provided values)
camera_matrix = np.array([
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
[959.7808651839981, 0, 298.2425204757662],
[0, 961.3210947068818, 271.3019300782519],
[0, 0, 1]
]
dist_coeffs = np.array([0.1456080062940991, -0.3525662795219286,
0.008754005511518403, -0.001863066062528824, 0])

# Object points in the robot's coordinate system (assuming the object is on a flat
surface)
object_points = np.array([
[-0.5, -0.5, 0], # Top-left corner
[0.5, -0.5, 0], # Top-right corner
[0.5, 0.5, 0], # Bottom-right corner
[-0.5, 0.5, 0] # Bottom-left corner
], dtype=np.float32)

# Initialize ROS node
rospy.init_node('dobot_control_node', anonymous=True)

# ROS Service proxies
set_ptp_cmd_service = rospy.ServiceProxy('/DobotServer/SetPTPCmd',
SetPTPCmd)
set_suction_cup_service = rospy.ServiceProxy('/DobotServer/SetEndEffectorSuctionCup',
SetEndEffectorSuctionCup)

# Default middle coordinates for the camera frame (set to 0 as requested)
middle_x_robot = 0
middle_y_robot = 110

# Basket coordinates
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
basket_1_coords = (150, 150, 100)
basket_2_coords = (240, 0, 100)
basket_3_coords = (0, -180, 100) # Coordinates for the unknown objects
```

```
def limit_param(value, min_value, max_value):
    """Ensure the value is within the min and max range."""
    return max(min_value, min(value, max_value))
```

```
def SetPTPCmd(x, y, z, r=0, ptpMode=0):
    """Move Dobot to specified coordinates."""
    global set_ptp_cmd_service
    try:
        if x != 0 or y != 0: # Avoid minimal parameter output
            req = SetPTPCmdRequest()
            req.ptpMode = ptpMode
            req.x = x
            req.y = y
            req.z = z
            req.r = r
            set_ptp_cmd_service(req)
    except rospy.ServiceException as e:
        pass
```

```
def SetSuctionCup(enableCtrl, suck, isQueued=0):
    """Control the suction cup on the Dobot."""
    global set_suction_cup_service
    try:
        req = SetEndEffectorSuctionCupRequest()
        req.enableCtrl = enableCtrl
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
req.suck = suck
req.isQueued = isQueued
set_suction_cup_service(req)
except rospy.ServiceException as e:
    pass
```

```
def compute_homography():
    """Compute homography matrix based on the points selected."""
    global homography_matrix
    if len(points) == 4:
        homography_matrix, _ = cv.findHomography(points, object_points[:, :2])
    else:
        homography_matrix = None

def calculate_centroid():
    """Calculate the centroid of the selected points (quadrilateral)."""
    if len(points) == 4:
        centroid_x = int(sum(point[0][0] for point in points) / 4)
        centroid_y = int(sum(point[0][1] for point in points) / 4)
        return centroid_x, centroid_y
    return (0, 0)

def calculate_delta(middle_x, middle_y, centroid_x, centroid_y):
    """Calculate the delta values in pixel units."""
    delta_x_pixels = centroid_x - middle_x
    delta_y_pixels = centroid_y - middle_y
    return delta_x_pixels, delta_y_pixels

def apply_homography_to_centroid(centroid_x, centroid_y):
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

""""Transform the centroid coordinates from the camera frame to the robot's coordinate system,

considering all directional relationships and calibration references.""""

if homography_matrix is not None:

```
# Apply the homography to get robot coordinates from camera coordinates
```

```
camera_coords = np.array([[centroid_x, centroid_y]],  
dtype=np.float32).reshape(1, 1, 2)
```

```
robot_coords = cv.perspectiveTransform(camera_coords,  
homography_matrix)
```

```
x_robot, y_robot = robot_coords[0][0]
```

```
# Apply the coordinate transformations based on the directional relationship
```

```
transformed_x_robot = -y_robot # Camera Y- (up) → Robot X+
```

```
transformed_y_robot = x_robot # Camera X+ (right) → Robot Y+
```

```
# Apply the specific condition: Camera Y=360 should map to Robot X=0
```

```
y_camera_reference = 360
```

```
x_robot_offset = y_camera_reference - centroid_y
```

```
transformed_x_robot += x_robot_offset
```

```
# Apply scale factor to both axes
```

```
transformed_x_robot *= scale_factor
```

```
transformed_y_robot *= scale_factor
```

```
# Adjusting based on pickup area limits
```

```
transformed_x_robot = limit_param(transformed_x_robot, X_MIN, X_MAX)
```

```
transformed_y_robot = limit_param(transformed_y_robot, Y_MIN, Y_MAX)
```

```
# Debug prints
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritrik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print(f"Centroid X, Y: ({centroid_x}, {centroid_y})")
print(f"Robot Coordinates: X={transformed_x_robot},
Y={transformed_y_robot}")

return int(transformed_x_robot), int(transformed_y_robot)

return 0, 0

def classify_surface(frame):
    """Classify the surface based on feature matching with reference images."""
    # Load reference images for pale and textured surfaces
    reference_pale = cv.imread(r'/home/nada/Downloads/ref2.jpeg', 0)
    reference_textured = cv.imread(r'/home/nada/Downloads/ref1.jpeg', 0)

    # Convert the current frame to grayscale
    gray_frame = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)

    # Detect keypoints and descriptors
    sift = cv.SIFT_create()
    kp1, des1 = sift.detectAndCompute(gray_frame, None)

    # Feature matching with reference pale
    kp2, des2 = sift.detectAndCompute(reference_pale, None)
    matches_pale = match_features(des1, des2)

    # Feature matching with reference textured
    kp2, des2 = sift.detectAndCompute(reference_textured, None)
    matches_textured = match_features(des1, des2)

    # Determine the best match
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

```
if len(matches_textured) > len(matches_pale):
    return 'textured'
elif len(matches_pale) > len(matches_textured):
    return 'pale'
else:
    return 'unknown'

def match_features(des1, des2):
    """Match features between descriptors using BFMatcher with RANSAC."""
    bf = cv.BFMatcher(cv.NORM_L2, crossCheck=True)
    matches = bf.match(des1, des2)

    # Apply RANSAC to filter matches
    if len(matches) > 10: # Ensure we have enough matches to apply RANSAC
        src_pts = np.float32([des1[m.queryIdx].pt for m in matches]).reshape(-1, 2)
        dst_pts = np.float32([des2[m.trainIdx].pt for m in matches]).reshape(-1, 2)

        _, mask = cv.findHomography(src_pts, dst_pts, cv.RANSAC, 5.0)
        matches = [matches[i] for i in range(len(matches)) if mask[i]]

    return matches

def update_labels():
    """Update the labels with the latest coordinates, homography, and centroid in Tkinter."""

    global homography_matrix, current_centroid, middle_x_robot, middle_y_robot

    if len(points) == 4 and homography_matrix is not None:
        current_centroid = calculate_centroid()
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
x_robot, y_robot = apply_homography_to_centroid(current_centroid[0],
current_centroid[1])
```

```
if x_robot is not None and y_robot is not None:
```

```
    delta_x_pixels, delta_y_pixels = calculate_delta(middle_x_robot,
middle_y_robot, current_centroid[0], current_centroid[1])
```

```
    label_coords.config(text=f"Robot Coordinates: X={x_robot},
Y={y_robot}")
```

```
    label_homography.config(text=f"Homography
Matrix:\n{homography_matrix}")
```

```
    label_centroid.config(text=f"Centroid: X={current_centroid[0]},
Y={current_centroid[1]}")
```

```
    label_delta.config(text=f"Delta:  $\Delta X$ ={delta_x_pixels} pixels,
 $\Delta Y$ ={delta_y_pixels} pixels\n"
```

```
    f" $\Delta X$ _robot={x_robot-middle_x_robot} units,  $\Delta Y$ _robot={y_robot-
middle_y_robot} units")
```

```
else:
```

```
    reset_labels_to_na()
```

```
def reset_labels_to_na():
```

```
    """Reset all labels to display 'N/A'."""
```

```
    label_coords.config(text="Robot Coordinates: N/A")
```

```
    label_homography.config(text="Homography Matrix: N/A")
```

```
    label_centroid.config(text="Centroid: N/A")
```

```
    label_delta.config(text="Delta: N/A")
```

```
def draw_polygon(frame, points):
```

```
    """Draw a polygon based on the points that have been stored."""
```

```
    if points is not None and len(points) == 4: # Ensure it's a quadrilateral
```

```
        centroid_x, centroid_y = calculate_centroid()
```

```
        pts = np.array(points, dtype=np.int32).reshape((-1, 1, 2))
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
cv.polylines(frame, [pts], isClosed=True, color=(0, 255, 0), thickness=2)
# Draw the centroid as a dot
cv.circle(frame, (centroid_x, centroid_y), 5, (255, 0, 0), -1)
```

```
def update_points(initial_gray, gray, old_points):
    """Update the position of points using optical flow."""
    new_points, status, _ = cv.calcOpticalFlowPyrLK(initial_gray, gray, old_points,
    None)
    if new_points is not None and status is not None:
        good_new = new_points[status.flatten() == 1]
        good_old = old_points[status.flatten() == 1]
        return good_new.reshape(-1, 1, 2), good_old.reshape(-1, 1, 2)
    return old_points, old_points
```

```
def draw_rectangle_with_click(event, x, y, flags, param):
    global points, frame, tracking, initial_gray
    if event == cv.EVENT_LBUTTONDOWN:
        if len(points) < 4:
            points.append([x, y]) # Store points as lists, not tuples
            cv.circle(frame, (x, y), 3, (0, 255, 0), -1)
        if len(points) == 4:
            tracking = True
            initial_gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
            points = np.array(points, dtype=np.float32).reshape(-1, 1, 2) # Convert to
            NumPy array after collecting all points
            compute_homography()
            update_labels() # Update immediately after homography computation
```

```
def execute_pickup_and_place():
    """Execute a sequence of commands to pickup and place an object."""
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

if current_centroid is not None:

```
x_robot, y_robot = apply_homography_to_centroid(current_centroid[0],
current_centroid[1])
```

if x_robot is not None and y_robot is not None:

```
x_robot_scaled = x_robot * scale_factor
```

```
y_robot_scaled = y_robot * scale_factor
```

```
x_robot_scaled = int(limit_param(x_robot_scaled, X_MIN, X_MAX))
```

```
y_robot_scaled = int(limit_param(y_robot_scaled, Y_MIN, Y_MAX))
```

```
# Move to above box position
```

```
SetPTPCmd(x_robot_scaled, y_robot_scaled, 100)
```

```
rospy.sleep(2)
```

```
SetPTPCmd(x_robot_scaled, y_robot_scaled, 50)
```

```
rospy.sleep(2)
```

```
SetPTPCmd(x_robot_scaled, y_robot_scaled, -25)
```

```
rospy.sleep(4)
```

```
SetSuctionCup(True, True)
```

```
SetPTPCmd(x_robot_scaled, y_robot_scaled, 50)
```

```
rospy.sleep(2)
```

```
SetPTPCmd(x_robot_scaled, y_robot_scaled, 100)
```

```
rospy.sleep(2)
```

```
SetPTPCmd(0, 180, 100) # Safe coordinate to move the boxes without
triggering the alarm
```

```
rospy.sleep(2)
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Classify the surface
surface_type = classify_surface(frame)

# Decide which basket to move to
if surface_type == 'textured':
    SetPTPCmd(*basket_1_coords) # Move to basket 1
elif surface_type == 'pale':
    SetPTPCmd(*basket_2_coords) # Move to basket 2
else:
    SetPTPCmd(*basket_3_coords) # Move to basket 3 (unknown or trash)

rospy.sleep(2)

SetPTPCmd(basket_1_coords[0], basket_1_coords[1], 0)
SetSuctionCup(True, False)
rospy.sleep(2)

SetPTPCmd(basket_1_coords[0], basket_1_coords[1], 100)
rospy.sleep(2)

def update_center_point():
    """Update the robot's center point based on user input in Tkinter."""
    global middle_x_robot, middle_y_robot, scale_factor

    try:
        middle_x_robot = float(entry_robot_middle_x.get())
        middle_y_robot = float(entry_robot_middle_y.get())

        scale_factor = float(entry_scale_factor.get()) / 10 # Adjust scaling factor as
        necessary

    except ValueError:
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
pass # Removed print statement to reduce terminal output
```

```
def opencv_loop():
```

```
    global points, frame, tracking, initial_gray, root, last_update_time,
    current_centroid # Declare points and last_update_time as global
```

```
    cap = cv.VideoCapture(2)
```

```
    cap.set(cv.CAP_PROP_FRAME_WIDTH, 640)
```

```
    cap.set(cv.CAP_PROP_FRAME_HEIGHT, 480)
```

```
    cv.namedWindow("Frame")
```

```
    cv.setMouseCallback("Frame", draw_rectangle_with_click)
```

```
    while True:
```

```
        ret, original_frame = cap.read()
```

```
        if not ret:
```

```
            break
```

```
        frame = original_frame.copy()
```

```
        # Draw a dot at the middle of the frame
```

```
        height, width = frame.shape[:2]
```

```
        middle_frame_x = width // 2
```

```
        middle_frame_y = height // 2
```

```
        cv.circle(frame, (middle_frame_x, middle_frame_y), 5, (0, 0, 255), -1)
```

```
    if tracking and initial_gray is not None:
```

```
        gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
```

```
        points, _ = update_points(initial_gray, gray, points)
```

```
    if len(points) == 4:
```

```
        draw_polygon(frame, points)
```

```
        compute_homography()
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
update_labels() # Update Tkinter labels after computing homography

initial_gray = gray.copy() # Update the reference frame

cv.imshow("Frame", frame)

key = cv.waitKey(1) & 0xFF
if key == ord('q'):
    break
elif key == ord('r'):
    points = []
    tracking = False
    initial_gray = None

cap.release()
cv.destroyAllWindows()
root.quit() # Quit the Tkinter main loop

def tkinter_loop():
    global root, entry_robot_middle_x, entry_robot_middle_y, entry_scale_factor
    global label_coords, label_homography, label_centroid, label_delta

    root = tk.Tk()
    root.title("Homography and Coordinates Display")

    label_coords = tk.Label(root, text="Robot Coordinates: N/A", font=("Open
    Sans", 12))

    label_coords.grid(row=0, column=0, padx=10, pady=5)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
label_homography = tk.Label(root, text="Homography Matrix: N/A", font=("Open Sans", 12))
```

```
label_homography.grid(row=1, column=0, padx=10, pady=5)
```

```
label_centroid = tk.Label(root, text="Centroid: N/A", font=("Open Sans", 12))
```

```
label_centroid.grid(row=2, column=0, padx=10, pady=5)
```

```
label_delta = tk.Label(root, text="Delta: N/A", font=("Open Sans", 12))
```

```
label_delta.grid(row=3, column=0, padx=10, pady=5)
```

```
tk.Label(root, text="Robot Middle Point X:").grid(row=4, column=0, padx=10, pady=5, sticky="e")
```

```
entry_robot_middle_x = Entry(root)
```

```
entry_robot_middle_x.grid(row=4, column=1, padx=10, pady=5)
```

```
entry_robot_middle_x.insert(0, str(middle_x_robot))
```

```
tk.Label(root, text="Robot Middle Point Y:").grid(row=5, column=0, padx=10, pady=5, sticky="e")
```

```
entry_robot_middle_y = Entry(root)
```

```
entry_robot_middle_y.grid(row=5, column=1, padx=10, pady=5)
```

```
entry_robot_middle_y.insert(0, str(middle_y_robot))
```

```
tk.Label(root, text="Scale Factor:").grid(row=6, column=0, padx=10, pady=5, sticky="e")
```

```
entry_scale_factor = Entry(root)
```

```
entry_scale_factor.grid(row=6, column=1, padx=10, pady=5)
```

```
entry_scale_factor.insert(0, str(scale_factor))
```

```
button_update_center = tk.Button(root, text="Update Center Point", command=update_center_point)
```

```
button_update_center.grid(row=7, column=0, colspan=2, pady=10)
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
button_execute = tk.Button(root, text="Execute Pickup and Place",  
command=execute_pickup_and_place)
```

```
button_execute.grid(row=8, column=0, columnspan=2, pady=10)
```

```
root.mainloop()
```

```
if __name__ == "__main__":
```

```
    # Run OpenCV and Tkinter loops in parallel
```

```
    opencv_thread = threading.Thread(target=opencv_loop)
```

```
    opencv_thread.start()
```

```
    tkinter_loop()
```

```
    opencv_thread.join()
```





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 5. Realisasi Program PythonUI

```
import tkinter as tk

from tkinter import Toplevel, Text, Scrollbar, Canvas, Frame

import subprocess # Import the subprocess module

import re

class DobotControlUI:

    def __init__(self, master):

        self.master = master

        self.master.title("Dobot Magician Tools UI")

        self.master.protocol("WM_DELETE_WINDOW", self.on_closing)

        self.master.geometry("480x800") # Set initial window size

        self.roscore_process = None

        self.dobotserver_process = None

        self.create_widgets()

    def create_widgets(self):

        # Create a canvas

        self.canvas = Canvas(self.master)

        self.canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

        # Add a scrollbar linked to the canvas

        self.scrollbar = Scrollbar(self.master, orient="vertical",

command=self.canvas.yview)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

self.canvas.configure(yscrollcommand=self.scrollbar.set)

# Create a frame inside the canvas to hold all the widgets

self.scrollable_frame = Frame(self.canvas)

self.canvas.create_window((0, 0), window=self.scrollable_frame,
anchor="nw")

# Bind the scrolling event to the canvas

self.scrollable_frame.bind("<Configure>", lambda e:
self.canvas.configure(scrollregion=self.canvas.bbox("all")))

# PTP Common Parameters Setup

ptp_frame = tk.LabelFrame(self.scrollable_frame, text="PTP Common
Parameters")

ptp_frame.pack(pady=5, fill=tk.X, expand=True)

tk.Label(ptp_frame, text="PTP Velocity Ratio (0 to 100):").grid(row=0,
column=0, padx=10, pady=5, sticky="w")

self.ptp_velocity_var = tk.DoubleVar(value=50)

tk.Entry(ptp_frame, textvariable=self.ptp_velocity_var).grid(row=0,
column=1, padx=10, pady=5)

tk.Label(ptp_frame, text="PTP Acceleration Ratio (0 to 100):").grid(row=1,
column=0, padx=10, pady=5, sticky="w")

self.ptp_acceleration_var = tk.DoubleVar(value=50)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
tk.Entry(ptp_frame, textvariable=self.ptp_acceleration_var).grid(row=1,
column=1, padx=10, pady=5)
```

```
set_ptp_button = tk.Button(ptp_frame, text="Set PTP Params",
command=self.set_ptp_params)
```

```
set_ptp_button.grid(row=2, column=1, pady=10)
```

```
self.ptp_params_label = tk.Label(ptp_frame, text="Current PTP Params: Not
Set")
```

```
self.ptp_params_label.grid(row=3, column=0, columnspan=2, padx=10,
pady=10)
```

```
# Setup for PTP Jump Parameters
```

```
jump_frame = tk.LabelFrame(self.scrollable_frame, text="PTP Jump
Parameters")
```

```
jump_frame.pack(pady=5, fill=tk.X, expand=True)
```

```
tk.Label(jump_frame, text="Jump Height:").grid(row=0, column=0, padx=10,
pady=5, sticky="w")
```

```
self.jump_height_entry = tk.Entry(jump_frame)
```

```
self.jump_height_entry.grid(row=0, column=1, padx=10, pady=5)
```

```
tk.Label(jump_frame, text="Z Limit:").grid(row=1, column=0, padx=10,
pady=5, sticky="w")
```

```
self.z_limit_entry = tk.Entry(jump_frame)
```

```
self.z_limit_entry.grid(row=1, column=1, padx=10, pady=5)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.ptp_jump_btn = tk.Button(jump_frame, text="Set Jump Params",  
command=self.set_ptp_jump_params)
```

```
self.ptp_jump_btn.grid(row=1, column=2, padx=10, pady=5, sticky="ew")
```

```
# Dobot Server, Suction Cup, and Alarm Controls
```

```
dobot_frame = tk.LabelFrame(self.scrollable_frame, text="Dobot Server  
Controls")
```

```
dobot_frame.pack(pady=5, fill=tk.X, expand=True)
```

```
self.tty_label = tk.Label(dobot_frame, text="Enter USB port (e.g.,  
ttyCH343USB0):")
```

```
self.tty_label.grid(row=0, column=0, padx=10, pady=5, sticky="w")
```

```
self.tty_entry = tk.Entry(dobot_frame)
```

```
self.tty_entry.insert(0, "ttyCH343USB0")
```

```
self.tty_entry.grid(row=0, column=1, padx=10, pady=5)
```

```
self.dobotserver_btn = tk.Button(dobot_frame, text="Run DobotServer",  
command=self.run_dobotserver)
```

```
self.dobotserver_btn.grid(row=1, column=0, padx=10, pady=5, sticky="ew")
```

```
self.stopservice_btn = tk.Button(dobot_frame, text="Stop DobotServer",  
command=self.stop_dobotservice)
```

```
self.stopservice_btn.grid(row=1, column=1, padx=10, pady=5, sticky="ew")
```

```
self.activate_suctioncup_btn = tk.Button(dobot_frame, text="Activate Suction  
Cup", command=self.activate_suctioncup)
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.activate_suctioncup_btn.grid(row=2, column=0, padx=10, pady=5,
sticky="ew")
```

```
self.deactivate_suctioncup_btn = tk.Button(dobot_frame, text="Deactivate
Suction Cup", command=self.deactivate_suctioncup)
```

```
self.deactivate_suctioncup_btn.grid(row=2, column=1, padx=10, pady=5,
sticky="ew")
```

```
self.get_alarm_status_btn = tk.Button(dobot_frame, text="Get Alarm Status",
command=self.get_alarm_status)
```

```
self.get_alarm_status_btn.grid(row=3, column=0, padx=10, pady=5,
sticky="ew")
```

```
self.clear_alarm_status_btn = tk.Button(dobot_frame, text="Clear Alarm
Status", command=self.clear_alarm_status)
```

```
self.clear_alarm_status_btn.grid(row=3, column=1, padx=10, pady=5,
sticky="ew")
```

```
self.emergency_stop_btn = tk.Button(dobot_frame, text="Emergency Stop",
bg="red", fg="white", command=self.emergency_stop)
```

```
self.emergency_stop_btn.grid(row=4, column=0, colspan=2, padx=10,
pady=5, sticky="ew")
```

Roscore Button

```
self.roscore_btn = tk.Button(self.scrollable_frame, text="Start Roscore",
bg='purple', fg='white', command=self.start_roscore)
```

```
self.roscore_btn.pack(pady=5, fill=tk.X)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Queue Frame
queue_frame = tk.LabelFrame(self.scrollable_frame, text="Queue Controls")
queue_frame.pack(pady=10, fill=tk.X, expand=True)

self.start_exec_btn = tk.Button(queue_frame, text="Start Execution",
command=self.start_execution)

self.start_exec_btn.grid(row=0, column=0, padx=10, pady=5, sticky="ew")

self.stop_exec_btn = tk.Button(queue_frame, text="Stop Execution",
command=self.stop_execution)

self.stop_exec_btn.grid(row=0, column=1, padx=10, pady=5, sticky="ew")

self.clear_queue_btn = tk.Button(queue_frame, text="Clear Queue",
command=self.clear_queue)

self.clear_queue_btn.grid(row=0, column=2, padx=10, pady=5, sticky="ew")

# Home and Watch buttons stacked
button_frame = tk.Frame(self.scrollable_frame)
button_frame.pack(pady=5, fill=tk.X)

self.home_position_btn = tk.Button(button_frame, text="Home Position",
command=self.set_home_position)

self.home_position_btn.pack(side=tk.LEFT, padx=5, fill=tk.X, expand=True)

self.watch_pose_btn = tk.Button(button_frame, text="Watch Get Pose",
command=self.toggle_watch_pose)

self.watch_pose_btn.pack(side=tk.LEFT, padx=5, fill=tk.X, expand=True)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.watch_pose_active = False # Flag to track if automatic GetPose is on

# Start watching pose if needed

self.watch_pose()

# Labels for Pose Data

self.pose_label = tk.Label(self.scrollable_frame, text="Current Pose:",
font=("Helvetica", 14))

self.pose_label.pack(pady=5, fill=tk.X)

self.pose_frame = tk.Frame(self.scrollable_frame)
self.pose_frame.pack(pady=5, fill=tk.X)

self.x_label = tk.Label(self.pose_frame, text="X: N/A", fg="red",
font=("Helvetica", 12))
self.x_label.grid(row=0, column=0, padx=10, pady=5, sticky="ew")

self.y_label = tk.Label(self.pose_frame, text="Y: N/A", fg="green",
font=("Helvetica", 12))
self.y_label.grid(row=0, column=1, padx=10, pady=5, sticky="ew")

self.z_label = tk.Label(self.pose_frame, text="Z: N/A", fg="blue",
font=("Helvetica", 12))
self.z_label.grid(row=1, column=0, padx=10, pady=2, sticky="ew")

self.r_label = tk.Label(self.pose_frame, text="R: N/A", fg="purple",
font=("Helvetica", 12))
self.r_label.grid(row=1, column=1, padx=10, pady=2, sticky="ew")
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.joint_label = tk.Label(self.pose_frame, text="Joint Angles: N/A",
fg="orange", font=("Helvetica", 12))

self.joint_label.grid(row=2, column=0, columnspan=2, padx=10, pady=2,
sticky="ew")

# Start watching pose periodically
self.watch_pose()

def show_terminal_output(self):
# Create a new top-level window
terminal_window = Toplevel(self.master)
terminal_window.title("Terminal Output")

# Add a text box with a scrollbar
text_frame = tk.Frame(terminal_window)
text_frame.pack(fill=tk.BOTH, expand=True)

scrollbar = Scrollbar(text_frame)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

self.text_output = Text(text_frame, wrap=tk.WORD,
yscrollcommand=scrollbar.set)

self.text_output.pack(fill=tk.BOTH, expand=True)
scrollbar.config(command=self.text_output.yview)

# Add a button to show the terminal output
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

```
show_button = tk.Button(terminal_window, text="Show Terminal Output",  
command=self.display_terminal_output)
```

```
show_button.pack(pady=5)
```

```
def display_terminal_output(self):
```

```
# Example of how to show output in the text box
```

```
output = "Example output text...\n" # Replace with actual output
```

```
self.text_output.insert(tk.END, output)
```

```
self.text_output.see(tk.END) # Scroll to the end of the text
```

```
def set_ptp_params(self):
```

```
velocity_ratio = self.ptp_velocity_var.get()
```

```
acceleration_ratio = self.ptp_acceleration_var.get()
```

```
# Construct the command packet
```

```
header = b'\xAA\xAA'
```

```
ctrl = 83 # Control ID for SetPTPCommonParams
```

```
params = struct.pack('<ff', velocity_ratio, acceleration_ratio) # Assuming float
```

```
for ratios
```

```
isQueued = 0 # 0 for not queued, 1 for queued
```

```
rw = 0 # 0 for write
```

```
length = len(params) + 4 # Control, rw, isQueued, and checksum
```

```
# Construct the packet
```

```
packet = struct.pack('<2sBBffBB', header, ctrl, rw, isQueued, velocity_ratio,  
acceleration_ratio, 0, 0)
```

```
checksum = sum(packet) & 0xFF
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

tanpa izin Politeknik Negeri Jakarta

```
packet += struct.pack('<B', checksum)
```

```
# Example command, replace with actual ROS service call to send packet
```

```
print(f"Sending packet: {packet.hex()}")
```

```
# After sending, fetch the current parameters and update the label
```

```
self.update_ptp_params()
```

```
def update_ptp_params(self):
```

```
# Example command, replace with actual ROS service call to get current  
params
```

```
# Simulated response for demonstration
```

```
response_packet = b'\xAA\xAA\x83\x00\x01\x32\x00\x00\x00\x00' # Sample  
response packet
```

```
result = struct.unpack('<2sBBfB', response_packet)
```

```
velocity_ratio = result[3] # Parsed velocity ratio from response
```

```
acceleration_ratio = result[4] # Parsed acceleration ratio from response
```

```
self.ptp_params_label.config(text=f"Current PTP Params: Velocity  
Ratio={velocity_ratio}, Acceleration Ratio={acceleration_ratio}")
```

```
def set_ptp_jump_params(self):
```

```
jump_height = float(self.jump_height_entry.get())
```

```
z_limit = float(self.z_limit_entry.get())
```

```
# Assume isQueued is desired for this operation
```

```
try:
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
subprocess.run(['rosservice', 'call', '/DobotServer/SetPTPJumpParams',
str(jump_height), str(z_limit), '1'], check=True)

print("PTP Jump Parameters set successfully")

except subprocess.CalledProcessError as e:

print(f"Failed to set PTP Jump Parameters: {e}")

except Exception as e:

print(f"Unexpected error: {e}")

def start_roscore(self):
try:

self.roscore_process = subprocess.Popen(['roscore'])

print("Roscore started")

except Exception as e:

print(f"Failed to start roscore: {e}")

def run_dobotserver(self):

tty_port = self.tty_entry.get()

try:

self.dobotserver_process = subprocess.Popen(['roslaunch', 'dobot',
'DobotServer', tty_port])

self.dobotserver_btn.config(bg="green") # Change button color to green

print(f"DobotServer started on {tty_port}")

except Exception as e:

self.dobotserver_btn.config(bg="red") # Ensure button color is red on
failure

print(f"Failed to run DobotServer on {tty_port}: {e}")
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def stop_dobotserver(self):
```

```
    try:
```

```
        if self.dobotserver_process:
```

```
            self.dobotserver_process.terminate()
```

```
            self.dobotserver_process.wait()
```

```
            self.dobotserver_btn.config(bg="red") # Change button color to red
```

```
            print("DobotServer stopped")
```

```
        except Exception as e:
```

```
            print(f"Failed to stop DobotServer: {e}")
```

```
def activate_suctioncup(self):
```

```
    try:
```

```
        command = "rosservice call /DobotServer/SetEndEffectorSuctionCup
```

```
'enableCtrl: 1\nsuck: 1'"
```

```
        subprocess.run(command, shell=True, check=True)
```

```
        print("Suction cup activated")
```

```
    except subprocess.CalledProcessError as e:
```

```
        print(f"Failed to activate suction cup, subprocess error: {e}")
```

```
    except Exception as e:
```

```
        print(f"Unexpected error: {e}")
```

```
def deactivate_suctioncup(self):
```

```
    try:
```

```
        command = "rosservice call /DobotServer/SetEndEffectorSuctionCup
```

```
'enableCtrl: 1\nsuck: 0'"
```

```
        subprocess.run(command, shell=True, check=True)
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

tanpa izin Politeknik Negeri Jakarta

```
print("Suction cup deactivated")
except subprocess.CalledProcessError as e:
    print(f"Failed to deactivate suction cup, subprocess error: {e}")
except Exception as e:
    print(f"Unexpected error: {e}")

def get_alarm_status(self):
    try:
        result = subprocess.check_output(['rosservice', 'call',
'/DobotServer/GetAlarmsState', '{}']).decode('utf-8')
        print("Alarm status: ", result)
    except Exception as e:
        print(f"Failed to get alarm status: {e}")

def clear_alarm_status(self):
    try:
        result = subprocess.check_output(['rosservice', 'call',
'/DobotServer/ClearAllAlarmsState', '{}']).decode('utf-8')
        print("Clear alarm status: ", result)
    except Exception as e:
        print(f"Failed to clear alarm status: {e}")

def emergency_stop(self, event=None): # Allow calling from button and
keyboard
    try:
        subprocess.run(['rosservice', 'call',
'/DobotServer/SetQueuedCmdForceStopExec'], check=True)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

tanpa izin Politeknik Negeri Jakarta

```
print("Emergency stop activated")
except subprocess.CalledProcessError as e:
    print(f"Failed to execute emergency stop, subprocess error: {e.stderr}")
except Exception as e:
    print(f"Unexpected error during emergency stop: {e}")

def start_execution(self):
    try:
        subprocess.run(['rosservice', 'call',
'/DobotServer/SetQueuedCmdStartExec'], check=True)
        print("Started execution of queued commands")
    except subprocess.CalledProcessError as e:
        print(f"Failed to start execution, subprocess error: {e.stderr}")
    except Exception as e:
        print(f"Unexpected error during start execution: {e}")

def stop_execution(self):
    try:
        subprocess.run(['rosservice', 'call',
'/DobotServer/SetQueuedCmdStopExec'], check=True)
        print("Stopped execution of queued commands")
    except subprocess.CalledProcessError as e:
        print(f"Failed to stop execution, subprocess error: {e.stderr}")
    except Exception as e:
        print(f"Unexpected error during stop execution: {e}")
```





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def clear_queue(self):
    try:
        subprocess.run(['rosservice', 'call', '/DobotServer/SetQueuedCmdClear'],
            check=True)
        print("Cleared command queue")
    except subprocess.CalledProcessError as e:
        print(f"Failed to clear queue, subprocess error: {e.stderr}")
    except Exception as e:
        print(f"Unexpected error during queue clearance: {e}")

def toggle_watch_pose(self):
    if self.watch_pose_active:
        self.watch_pose_active = False
        self.watch_pose_btn.config(text="Watch Get Pose")
        print("Automatic GetPose deactivated")
    else:
        self.watch_pose_active = True
        self.watch_pose_btn.config(text="Stop Watching Pose")
        print("Automatic GetPose activated")
        self.watch_pose()

def watch_pose(self):
    if self.watch_pose_active:
        try:
            result = subprocess.check_output(['rosservice', 'call',
                '/DobotServer/GetPose', '{}']).decode('utf-8')
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
if not result.strip():
    raise ValueError("Empty response from rosservice")

pose = self.parse_pose(result)

self.x_label.config(text=f"X: {pose['x']}")
self.y_label.config(text=f"Y: {pose['y']}")
self.z_label.config(text=f"Z: {pose['z']}")
self.r_label.config(text=f"R: {pose['r']}")
joint_angles = ', '.join([str(angle) for angle in pose['jointAngle']])
self.joint_label.config(text=f"Joint Angles: {joint_angles}")

print("Pose: ", pose)
except ValueError as ve:
    print(f"Failed to get pose: {ve}")
except Exception as e:
    print(f"Failed to get pose: {e}")
finally:
    # Schedule the next call to this method after 1.8 seconds (1800
milliseconds)

self.master.after(1800, self.watch_pose)
```

```
def parse_pose(self, result):
```

```
    pose = {
        'x': 0.0,
        'y': 0.0,
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
'z': 0.0,
'r': 0.0,
'jointAngle': [0.0, 0.0, 0.0, 0.0]
}

# Use regular expressions to find the values
pose['x'] = float(re.search(r"x: ([\d\.-]+)", result).group(1))
pose['y'] = float(re.search(r"y: ([\d\.-]+)", result).group(1))
pose['z'] = float(re.search(r"z: ([\d\.-]+)", result).group(1))
pose['r'] = float(re.search(r"r: ([\d\.-]+)", result).group(1))
joint_angle_match = re.search(r"jointAngle: \\.([\^]+)\.", result)
if joint_angle_match:
    pose['jointAngle'] = [float(val) for val in
joint_angle_match.group(1).split(',')]
return pose

def set_home_position(self):
try:
    subprocess.Popen(['rosservice', 'call', '/DobotServer/SetHOMECmd'])
    print("Home position set")
except Exception as e:
    print(f"Failed to set home position: {e}")

def on_closing(self):
    self.stop_dobotserver()
    self.stop_rosscore()
    self.master.destroy()
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def stop_roscore(self):  
    try:  
        if self.roscore_process:  
            self.roscore_process.terminate()  
            self.roscore_process.wait()  
            print("Roscore stopped")  
    except Exception as e:  
        print(f"Failed to stop roscore: {e}")  
  
if __name__ == "__main__":  
    root = tk.Tk()  
    app = DobotControlUI(root)  
    root.mainloop()
```





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

tanpa izin Politeknik Negeri Jakarta

Lampiran 6. Realisasi Program Homografi

```
import cv2
import numpy as np

# Initialize variables
points = []
tracking = False # Status for determining whether tracking is active
initial_gray = None # Store the initial grayscale frame for optical flow
selecting = False # Status to control when selection starts

def update_points(initial_gray, gray, old_points):
    """Update point positions using optical flow."""
    new_points, status, errors = cv2.calcOpticalFlowPyrLK(initial_gray, gray,
old_points, None)
    if new_points is not None and status is not None and len(status) ==
old_points.shape[0]:
        good_new = new_points[status.flatten() == 1]
        good_old = old_points[status.flatten() == 1]
        return good_new, good_old
    else:
        return None, None

def update_feature_points(frame, old_points, old_gray):
    """Update points using feature detection and matching."""
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    detector = cv2.ORB_create()
    kp1, des1 = detector.detectAndCompute(old_gray, None)
    kp2, des2 = detector.detectAndCompute(gray, None)

    # Match features using FLANN
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
matcher =
cv2.DescriptorMatcher_create(cv2.DescriptorMatcher_FLANNBASED)

matches = matcher.match(des1, des2)

matches = sorted(matches, key=lambda x: x.distance)

# Update points based on matching
new_points = np.array([kp2[m.trainIdx].pt for m in matches],
dtype=np.float32).reshape(-1, 1, 2)

return new_points, gray

def draw_polygon(frame, points):
    """Function to draw a polygon based on saved points."""
    if points is not None and len(points) > 1:
        print("Drawing polygon with points:", points) # Log points
        pts = np.array(points, dtype=np.int32).reshape((-1, 1, 2))
        cv2.polylines(frame, [pts], isClosed=True, color=(0, 255, 0), thickness=2)
    else:
        print("Not enough points to draw a polygon.")

def draw_rectangle_with_click(event, x, y, flags, param):
    global points, frame, tracking, initial_gray, selecting
    if selecting:
        if event == cv2.EVENT_LBUTTONDOWN:
            if len(points) < 4:
                points.append((x, y))
                cv2.circle(frame, (x, y), 3, (0, 255, 0), -1)
            if len(points) == 4 and not tracking:
                tracking = True
                selecting = False # Stop selection after 4 points are selected
                initial_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
points = np.array(points, dtype=np.float32).reshape(-1, 1, 2)
```

```
cap = cv2.VideoCapture(0)
```

```
cv2.namedWindow("Frame")
```

```
cv2.setMouseCallback("Frame", draw_rectangle_with_click)
```

```
while True:
```

```
    ret, original_frame = cap.read()
```

```
    if not ret:
```

```
        break
```

```
    frame = original_frame.copy()
```

```
    if tracking and initial_gray is not None:
```

```
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
        new_points, _ = update_points(initial_gray, gray, points)
```

```
        if new_points is not None:
```

```
            points = new_points
```

```
            draw_polygon(frame, points)
```

```
        else:
```

```
            points, initial_gray = update_feature_points(frame, points, initial_gray)
```

```
            if points is None or len(points) < 4:
```

```
                tracking = False # Reset tracking if points cannot be recovered
```

```
            initial_gray = gray.copy() # Update reference frame
```

```
cv2.imshow("Frame", frame)
```

```
key = cv2.waitKey(1) & 0xFF
```

```
if key == ord('q'):
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
break
```

```
elif key == ord('i'):
```

```
    points = [] # Reset points
```

```
    tracking = False
```

```
    selecting = True
```

```
    print("Please click on 4 points of the object.")
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```



Lampiran 7. Link Program Pemetaan Dobot Magician

<https://github.com/doniharley/dobotMarucomapping>



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Lampiran 8. Link Program homography Dobot Magician

<https://github.com/doniharley/DobotHomography>



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

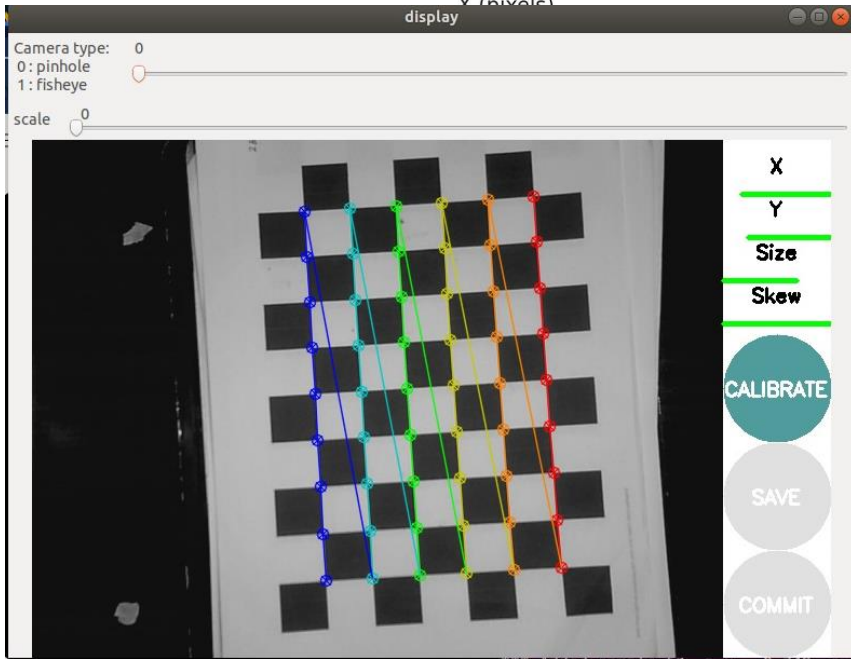
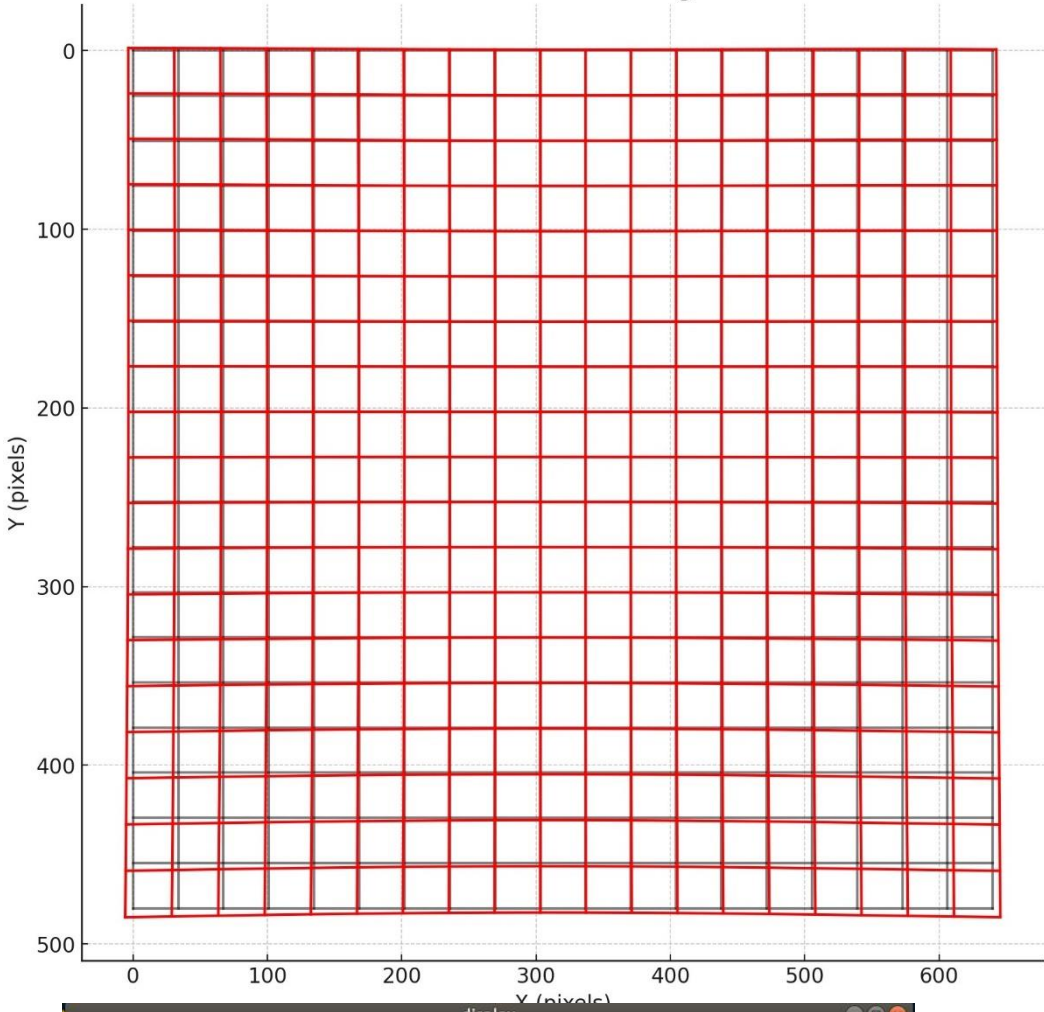
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Lampiran 9. Alat bantu Kalibrasi Kamera

Distortion Effect on Image Grid



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta